GitOps with Azure Kubernetes using ArgoCD

Problem Statement

Kubernetes is the Industry standard today to run application workloads with its orchestration capabilities like self healing, auto scaling etc.

Azure Kubernetes Service (AKS) is a **managed Kubernetes service** that allows users to deploy and manage containerized applications on Azure.

For every application running on Kubernetes platform, there are couple of components. One is **application source code** and other one is related to **cluster management.**

Continuous Integration part covers the **application** and **Continuous Delivery** part covers the **Cluster management**.

As part of **Continuous Integration**, when application source code commits in GitHub, automated CI pipeline pulls the code, builds, run test cases, generate Docker file and upload to Container registry. In addition, it updates the Kubernetes configuration file (deployment.yaml) with the latest Docker image version.

Continuous Delivery pipeline, deploys the updated Configuration files (deployment.yaml, service.yaml and others like configmap, secrets, namespace etc else helm charts) on Kubernetes Cluster. This is more of **push based approach** where the configuration files are applied on target Cluster with **kubectl apply** command.

Opportunity for improvement in this approach (non GitOps way of CICD) as follows:

- Even when there is no change in application code like updating replicas in deployment.yaml scenario as well, we have to execute both CI and CD jobs.
- **Cluster credentials to be stored in a vault** for CICD process to leverage and push changes. This is a **security risk** situation
- In Kubernetes deployments, we *can't check for the successful completion of deployment*. Only we can execute the Kubectl apply command as part of deployment. We may need to execute test scripts to validate the completion of deployment

These areas of improvement items are handled in GitOps way of working enabled by tools like Argo CD, Flux. In this blog, we will be covering the Argo CD way of implementation of GitOps in Azure Kubernetes.

Azure Kubernetes come with "GitOps" feature leveraging Flux as its feature. Hence, I'm more Intrested in covering the "Argo CD" way of implementation. Audience of this blog may review "Flux" GitOps implementation provided by Azure Kubernetes which is much simpler. A key information to note - **Argo CD is not a replacement for CD tools like Jenkins, Azure DevOps release pipelines** etc. Its an enabler for GitOps specifically for Kubernetes platform. Argo CD is a CD tool. CI process may continue to get executed in AS-IS way. Argo CD to apply the changes from manifest files. Hence, its a CD tool.

Solution/Architecture

What is GitOps ?

GitOps is a way of implementing continuous deployment for cloud native applications, using git as a single source of truth for declarative infrastructure and applications. In a GitOps workflow, the desired state of an application and its infrastructure is stored in a git repository, and any changes to this desired state are made by committing changes to the git repository. An automated system, such as a Kubernetes operator or a continuous deployment tool, is then used to monitor the git repository and apply any changes to the actual state of the application and infrastructure.

GitOps benefits -

- Ability to track changes to the application and infrastructure, roll back changes easily, and collaborate with other team members using standard Git workflows.
- Allows for separation of concerns between development and operations teams, as the application and infrastructure configuration can be managed independently.

To implement GitOps in Azure Kubernetes, you will need to follow these steps:

- 1. Create a Kubernetes cluster on Azure using AKS.
- 2. Install and configure Argo CD on your AKS cluster.
- 3. Connect Argo CD to your Git repository, where you will store your declarative infrastructure and application configuration.

4. Define your application and infrastructure using Kubernetes manifest files and commit them to your Git repository.

5. Argo CD will automatically sync the desired state of your application and infrastructure with the actual state in your AKS cluster.

Argo CD is a pull based tool which picks the changes form GitHub rather pushing the code with kubectl commands

Technical Details and Implementation of solution

In order to implement GitOps with Argo CD in Azure Kuberentes, follow the below steps

← → C (i github.com/RAMAVEDA/k8s-argood-deployment								
💭 Samhar, my Ir								
B RAMAVECA /k8s-argood-deployment (mail)	R4MAVEDA /k8s-ergocd-deployment (
O Code (c) Issues (c) Pull verands (c) Article (t+	Projects III Wild () Sciently \ge Insights () Soft	17						
	Princip Princip Streps	Bollo ile Additiere Ci Codere						
	🐣 RAMAVEDA Update application yard	ecoclus, 13 minutes ago 🛞 A commita						
	14 modulation gent Spatia expedication	ngant Winnsteinge						
	19 - Ministry and Cost Straphy and	diyan ti Minanatas aya						
	(† samensyan) Costo serviceja							
	Heppeople interveted in this repository understand your proje	ict by adding a READVE AND a READVE						

1. Setup GitHub

Create **deployment.yaml**, **service.yaml**, **application.yaml** files as noted below in GitHub repo



deployment.yaml - Configuration file contain the container information like Docker image, replicas, resource limit etc. This defines the container information for Kuberenetes to orchestrate Make a note in the below image, we have "2" replicas. Hence, when we deploy, we expect 2 PODS of NGINX to get deployed

service.yaml - Configuration file to expose the pod/container via Load Balancer service so that it can be accessed from Browser



application.yaml - Configuration file specifically for Argo CD implementation which is a bridge to integrate Argo CD with Git by configuring the path of GitHub repo, path of config files, etc.



In this case, we have used "**nginx**" docker image for deployment.

Page 4 of 15

In full blown production size applications, we may need to have couple of repositories. One for hosting the application source code and other repository for hosting the configuration files for Cluster management. In this demonstration, we focus on Cluster management (Continuous Delivery part) with Argo CD in AKS

2. Create Azure Kubernetes Cluster

In order to create a Azure Kubernetes Cluster, we may need to have a subscription and resource group available as pre-requisite.

You may search for "Kubernetes" in the search of services and click create Kuberentes Cluster as noted below:



Upon click of "Create Kubernetes Cluster", below form will open.

In the below image, you may absorb, we are creating the resource group as part of Cluster creation process. Also, you may need to provide the name for Cluster as well.

Post filling the form (across all tabs), you may click "Review and create"

♦ ⇒ C (# printer	uncom/icrash/Microsoft.435			в.	a d	- sir	e.	*	ч,	
- MENORTATION		# torb record, error or	Period Sector		63	6 O	6	Φ	Ψ.	
Create Kebernetes clus	Gar									
Emris Michaevin Amas H Assertidaemin Similari adalah sebagai tertaka meneraka adalah sebagai sebagai tertaka Jam nen dalah adalah sebagai tertaka Pagat Baha Santa sebagai nenanga tertaka	device streature Alexand 5 per state fiscance enhancement while to a orderation would be alexand without the orderation of the streat without the state of the streat state of the streat streats and rest are mount propriate the	a Parier + crans intention: a fight; and temps off of modes stratics and no application of the anna regular and mesope is put								Post
handster" D Anaceptor" D	Yaad Badalara ke kisulatan Badaran kisadaray Satu tur	V V								1030
Onder Brails Distorprodisprigoation	Earlief (2) in good contract part to be here control of configurations along Along and workly. Non-con- agent control of compared match.	to Igentione of the power Igentione at any time								
References distances * 🗧	and Repairs dance	7	 The ranks are extracted or participations wenders underscards, and highers the tasks must shart and and and with a start is summary. Conserves werks merits are must invariantly the constitutions are summary. 							
Rejen* O	98 David K	~								
Autually some O	Ense GJ 1 Spt walkolig is recommended in souch	configuration.								
Name and Art State	1211/(default)	·Ψ.								
Wisemanhölle 5	 Belance Bedraiss for weighting: C. Male 									
Andrew A Lance	from Berli, Finit wolk +									

Validation pass, you may click "create" button to create the cluster as noted below:

← → C (i potsi.	zzane.com/incelate/usionole/LArca	医气白炎 萨弗平
 MountAses 	β^{2} Smooth transitions particular and then the ξ	23 23 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
Create Kubernetes cl	uster -	
Produce proved		
tanies Hasie you's derma	hemening integration determine Tage.	
lasire		
Julya, deriver	Yourd Stativ/Every for Infaulteties	
Amount pour	test war Weptimery	
Argim	Control	
Talamates aluatormente	aur Wegalar dater	
Eduarriches consum	138/10	
Loonatro preis	han.	
Subgert		
40.0x 10.04		
Indials of state services	Daniesi	
Annes		
Amount Meeting	International nanapolitions is	
and mustified	Indited	
Administration and Astronation	Losal masses with Calemanias NAC	
Interpolitan learn	Default Drugotter at weit with a plantere managed lev	
Interview.		
Network prefigeration	Edward	
245 sums parts	som Wingation during and	
	10.00	

AKS Cluster as noted below will be created. You may next connect to the cluster.



As noted in below image, you may connect to the cluster by clicking, "connect" button. Commands will be shown on the right side. You may use the "Cloud Shell" to execute the commands as noted below to connect to cluster and check the "default" deployments.

€ → C 🗎 portal.azu	ne.com/#@ramkivedagmail.orn	nicrosoft.com/resource/subscriptions/	56351c3-9605-42:9-b7a1-122a72412b3b/resourcegrou	8
= MinerfiAme		→ Search resources, emolans, and doub(Girl)		
Home > microsof.aki-20121218012419	Denies) aze-Noather-caste			
azure-blogathan-du	aster Namespaces			
,₽ Search <	+ Ceate 🗸 🔋 Delete 🗘 Refresh	🕼 thou labels 🖉 Give feedback		
· Outring	Filter by namespace same			
Acarbics	Etter he full namepaio name	(7 Add label filter)		
Access control UAM				
Ø Tan	Name Name	2 anu	Ape	
Marcan and advectories	kabe-sode-lease	Ative	6minutes	
Microsoft Defender for: Could	hbe-subic	O Artive	6-minutes	
Nobernetwareavercas	lake_vates	O Adver	Eminutus	
Samesees	ddad	Ative	6-minutes	
S Workloads				

In the below image, we can observe the default namespaces created by Azure Kubernetes when Cluster get created.

3. Install Argo CD in Azure Kuberenetes Cluster

1. Create a namespace in AKS cluster using the below noted command:

kubectl create namespace argocd



This namespace will hold all Argo CD related objects

2. Install Argo CD in AKS cluster using the below noted command:

kubectl apply -n argocd -f <u>https://raw.githubusercontent.com/argoproj/argo-cd/stable/</u> manifests/install.yaml



3. Change the argocd-server service type to LoadBalancer:

kubectl patch svc argocd-server -n argocd -p '{"spec": {"type": "LoadBalancer"}}'

4. Access Argo CD:

Now you will be able to see that the argocd-server service type has been changed to a LoadBalancer type. This means that it now has a public Azure load balancer attached to it with an external IP.

kubectl get svc -n argocd

Note: In production environments, use an ingress for the Argo CD API server that is secured. Above noted approach for exposing the Argo CD service is more for test/dev/ demo exercises.

Click the "external IP address" to launch Argo CD on browser as noted in below image:

$\in \ \Rightarrow \ \bigcirc \ ($ a potence	ura-com	NTPramkivadisgmail.or	microsoft.com/	elource, tubecrip	6.0ml/3843.6143-94656-4345	HE701-1	1251/22/1323	b/velourcegro	u. y e	9 (R D I	* =	🖬 🚯 (Update
 Mywikam 			/ Jenurous	ran, pervise, analytics (S	<i>.</i> #				1	8 G 🖗 8	0 R	saminisarianjaprasi nami bilan reservari analaria
NAME & DOCUMENTS OF A DESCRIPTION	. Janin	7 MURACINE CLEM										
azure-blogathan-cl	uster	Services and ingre										
11 Sealth 1	Service	Tores										
· Status	Eng th	anta sena la falta vas rena	Harbyroniagos Jil sanoparis	-	1							
S ADDRESS ADDR		Sets.		in the second	una ina		394	Down P	(energial of	ine.		Apr 1
P Transmittation without	T	Mecality		érimit.	0	04	Outoff'	11801		-0.90		T single
Monotorinania.Com	ŏ	Market State		halls app	un P	01	Out-P	1100/0		11,489,51,787		2 sinds
Colomour mourn		NALMANNE .		1004-00		08.	Culter	10067370		+4/10		T sought
Li surregion	0	examination of anterla		- ped		06	Gate#	1024710		1007109064,707		1 minute
S Wellinski		tradionator.		-9-2		08.	Galett	1000482		INV ONLY OF	106/109	1 800,05
A technicity care		Machinette:		agent			1 LUNITY	111110-001		40.000		1 10/0/10
Charter .		Fract with shore an other	NING .	-9-7	0	GR	Cubett	10361336		10217-024		1 Martin Par
E satisates		and the second s		apra	0		1 Lunerer	100.050.000		1.00.00		1 10/0/10
Address 1		even der seiner		apri .	5	08	Gamer	10038130		ADD TOPODATO		i minune
K Subandi		arooch samer		4013	0	UK .	LOCATION OF	10111-040	anni anni anni a	0.055.000.00	10.10	1 16/0/20
Entrantination		even i se se recito		apro i	5	OB.	Gamtr	102310		1010107		i minute
	s ↔ p	, • Mulweth prich ave :	orced-mover -	e ecessoi →e 'd'er	ee'i ("me'i "Koodelato	a ⁴¹⁵¹						- 0
same out apparture parters manufactures -7888-0.9100-0	e 104.81.9	BERNEL BER SHO -D	argoot	Termini -11	NAME IN							
arganismigi Lain I Maat John Vir Arganis din taaree Arganisminin Arganisminin Arganisminin Arganisminin	llier Genuner	Classer Classer Classer Classer Classer Classer Classer	10.0.10.00 10.0.10.00 10.0.00.00 10.0.00.00 10.0.00.00 10.0.00 10.0.00 1	COMIC COMIC COMIC COMIC COMIC COMIC COMIC	YOR TEP, AREANCE NER TEP, AREANCE NER TEP, AREANCE, AREAN NER TEP SERVICE SERV	8 1 5 5 5						
argood-aaroar argood-aaroar -aetrica aaarii in maa () -fidaray inar	andig (Clarball	16.0.131_343 16.0.5.50	12.194.204.99 Capital	04433027907,04342041472 04037228	07 59 48						

5. Credentials to access Argo CD:

Default User name: admin

Password can be obtained by executing the below noted command:

kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d; echo

Note: Warning message to launch Argo CD over browser may come. You may go-ahead and click the link to access the URL of Load balancer IP

6. Login to Argo CD as noted below:



Above noted image is the "Home" page of Argo CD. Since we are launching for the first time, no applications are listed.

4. Establish connection between Argo CD and GitHub



In order to establish connection between Argo CD and GitHub, we may need "application.yaml" to be applied in Cluster

Clone the Git repository and navigate the application.yaml as noted below and apply application.yaml using:

Kubectl apply -f application.yaml

Buch $\sim \odot ? \otimes \mathbb{D}_{r} ^{*} \odot \mathbb{B}$	
ramakrishman [~/k8e-nginz-confiç/k8e-argood-deployment]\$ is application.warideployment.wanicervice.yari	
<pre>ramakrishman (-/k8s-nyinx-config/k8s-argood-deployment)9 kubect1 apply -f application.yaml amplication.account.io/sairx_configured</pre>	
ramakrisknan (-/k8s-nginz-config/k8s-argoed-deployment)\$	

As expected, as we have defined 2 replicas in deployment.yaml, we have 2 PODS running in Cluster post sync



5. Argo CD synchronise the changes in GitHub configurations

Now, we are updating the replicas to "5" in deployment.yaml in GitHub and committing the change.





GitHub reference repository: https://github.com/RAMAVEDA/ k8s-argocd-deployment.git

Clean up:

As Best Practise, you may need to clean up the setup to avoid unexpected cost incurred. As noted below, its always advisable to delete the "Resource Group" to ensure no unnoticed services run

E \rightarrow CF (& persized	re een Wilremkivedapmeil.comicrose/t.cem/resource/tubse/artions/253351s2_03	05 42x0 67x1 122x72412x33/resourceOrau.	(Lodets : C () (Uodets
Maranda Auto	P. James measure services and desc(2m)		E S S S S S S Statute
Mana 4			
azure-biogathan-rg	P # -		:
P Search	de danke 🕼 Mangelske 🗤 😰 Mahamatane group 🔘 Mahah. 🛓 Egenter CH. 📆 Agence any	(0. Anigettep: 11 Mare -) (2) Data - § Equationples	D Open in mobile
N Baller	A Barrier Balan wasare yang		(BOS Aine
Contraction (Contraction)	Suburgetina (esse) + youri Socia, interneta Suna inter	inglighters - 2.5 contest	
5 accounts and	Schorpfise B YESSING AND LODGED WITH CON-	testiar	
∲ las	Ner add i Children is adding		
& teorescentes	hanna kanadalar		
F bath			
intege	The terms with the space of State in the second set (State in the secon		
L anona	Anating i mind i resprin. 🔲 Shar tidden gan. C		Magnaging w 20 Lin day v
9 Jaurile	Ober!	Terr 14	Lemine 1.
Q false		Education	formal info
# Incentio			11-12-02
ð ma			
Cost Menagement			
 Extended 			
🔍 Saturbacherini			
6 scoss			
Addate management and the			
Manitaling			
P Leidelander			
🕶 6849			
of mests			
intercal antire			
A 100			

Way forward:

Till now, we have experienced how Argo CD implements/syncronize the changes in Azure Kubernetes Cluster based on changes happen in GitHub repository. This is more focussed on Continuous Delivery. As an extension of this demo, we can include the application code for Continuous Integration in a separate repository and build a CICD pipeline with GitHub actions

Create a new GitHub Actions workflow for your repository. The workflow should include the following steps:

Continuous Integration to include -

- Checkout the code from your repository.
- Build and test your application.
- Push the updated application image to a container registry, such as Docker Hub or Azure Container Registry.
- Update the Kubernetes manifest files in your repository with the new image tag, and commit the changes to the repository.

Continuous Delivery to include the below step -

• Argo CD will automatically detect the changes and deploy the updated application to your AKS cluster.

In addition, Azure Kuberentes Cluster and Argo CD installation can be automated with Terraform for Cluster creation source code in Git as well

Challenges in implementing the solution

Creating the synchronisation between Argo CD and GitHub as the application.yaml specification have to be accurate for establishing the sync

Business Benefit

Below noted Business Benefits are key outcomes of GitOps way of implementation. However its not just limited to below benefits.

Faster Time to Market: With GitOps way of deployment automation, we have integrated feedback control loop which speeds up mean time to deployment. This increases the Time to Market which is essential to gain Market share

Higher Developers Utilization: Limited experience is suffice for developers with Kuberentes as Git way of management eases out the Cluster management and application deployment as Git and Argo CD controls both application and Cluster management. Developers can focus on application development rather diluting the effort on Kubernetes

Improved stability, consistency: As Cluster management is automated, stability naturally improves. Also as both application and cluster configurations are managed in Git as single source of truth, consistency and standardisation of applications improve significantly.

Higher reliability: Ability to rollback and fork, enable teams gain stable and reproducible rollbacks which improved the mean time to recovery (reduced from hours to minutes)

Higher Security for Organizations: As the credentials of Cluster is no longer managed outside of Cluster, security of Cluster improves which enables true "DevSecOps" way of implementations with GitOps for Kuberenetes platforms